[Kadu, 4(5): May, 2015]
ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

IMLEMENTATION OF SECURE DATA EXCHANGE PROTOCOL FOR P2PDSS IN PUBLIC HEALTH DOMAIN USING PAIRING-BASED CRYPTOGRAPHY

Amol G. Kadu, Prof. D. M. Dakhane

Department of computer science and engineering,sipna college of engineering and technology, India

ABSTRACT

Implimentation of novel secure data exchange protocol for P2PDSS in public health domain using pairing-based cryptography and data exchange policy between peers. In P2P eHealth data sharing scenarios, peers may need to exchange highly confidential data among them. Hence, there are some security threats that need to be considered using the protocol, any two peers that need to exchange data over an insecure medium can generate on-the-fly a secret session key by exchanging some system and session parameters. An important feature of the proposed protocol is that peers always generate a new session key for every new data exchange session; therefore, every session is completely independent with respect to the session key generation. The proposed protocol is robust against man-in-the middle attack, masquerade attack and the replay.

KEYWORDS :e Health, P2P, On-The-Fly Session Key, ECC.

INTRODUCTION

There is a lots of research concerning frameworks and mapping issues among peers, the aspect of sharing data between trusted peers in an anonymous and secured way is given less attention. Due to the security holes, P2PDSS is not being adopted in a practical scenario such as eHealth data sharing systems. A peer in a P2P Data Sharing System (P2PDSS) works as a client/server according to the policy of data exchange between the peers, and it is a highly scalable system. The local databases on peers are called peer databases. In P2PDSS, there is no global mediated schema like in the traditional data integration systems, where a global mediated schema is required for data exchange. There is an increasing interest in the creation of peer-to-peer database systems, which includes establishing and maintaining mappings between peers, processing queries using appropriate propagation techniques, and exchanging data between peers [3, 11, 12, 13, 14].

P2P systems are successfully used in several domains such as: file sharing, computing power sharing and instant message exchange. Due to their "good" features, new domains aim to take advantage of these systems. In the public health domain, for instance, we can cite some examples :(i) a doctor in a hospital may want to share most of his own data with other colleagues and to hide a portion of his data for personal reasons (e.g. data of an experience concerning a new drug for Alzheimer's disease); (ii) a doctor treating ill person may want to access the databases of the family doctor and the pharmacy of his patient in order to know his medical history and (iii) several researchers around the world are working on a drug for disease want to share data stored in their databases during an experience.

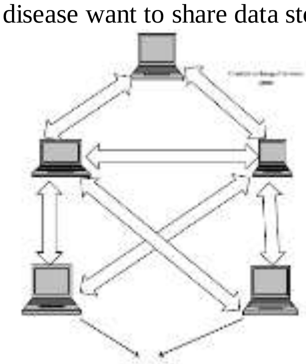


Fig 1. The peer to peer (p2p) architecture

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

The acquaintances between peers are established with predefined policies and trust relationships without having a centralized security policy. But, centralized-trusted control system is needed for the public key infrastructure (PKI). Therefore, the existing conventional PKI is not suitable to apply in e Health P2PDSS. Recent progress of Elliptic Curve Cryptography (ECC) [1], Identity-Based Cryptography (IBC) [3], and Pairing-based cryptography (PBC) [2] show that it is feasible to implement PBC on ECC. It have shown that ECC consumes considerably less resources than conventional public key cryptography (PKC) for a given security level [4].
In order to achieve secured data exchange in an eHealth P2PDSS dynamic network, this protocol based on Identity Based Encryption (IBE) and ECC. Using bilinear properties, each peer in the network generates a dynamic secret session key based on the attributes mentioned in the query and the predefined data exchange policy. In this protocol, peers authenticate each other in a pair-wise fashion without a centralized authentication policy. The protocol is mainly a secure session key generation for secure data exchange between peers. In brief, our protocol has the following properties: (1) flexible message-oriented secure data exchange between peers (2) exchange of data between peers without any third party certificates (3) communication between peers could be as simple as a single TCP connection (4) both parties (i.e. source and target) authenticate each other during data exchange.

Objectives of the Work

The main goal of the thesis work is to investigate security threads in *p2p data sharing system* that are raised in various existing file sharing systems in p2p network.
The overall objectives of my ME work are:
1) Analyzing and improving the peer to peer data sharing system ;
2) Analyzing various security threats in p2p data sharing system.
3) Analyze the possibilities various security measures to be taken for secure data exchange
4) Designing the Software Architecture, and
5) Developing a prototype for secured p2p data sharing system .

ANALYSIS OF PROBLEM.

As with software implementation today most P2P software is insecure. It is well known that the installation of this software create new methods for malicious users to cause damage
There is a lots of research concerning frameworks and mapping issues among peers, the aspect of sharing data between trusted peers in an anonymous and secured way is given less attention. Due to the security holes, P2PDSS is not being adopted in a practical scenario such as eHealth data sharing systems. A peer in a P2P Data Sharing System (P2PDSS) works as a client/server according to the policy of data exchange between the peers, and it is a highly scalable system.

SYSTEM DESIGN

In order to achieve secured data exchange in an eHealth P2PDSS dynamic network, in this project we presents a protocol based on ECC Based Encryption (IBE) and PBC. Using bilinear properties, each peer in the network generates a dynamic secret session key based on the attributes mentioned in the query and the predefined data exchange policy. In this protocol, peers authenticate each other in a pair-wise fashion without a centralized authentication policy. The protocol is mainly a query-based secure session key generation for secure data exchange between peers.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

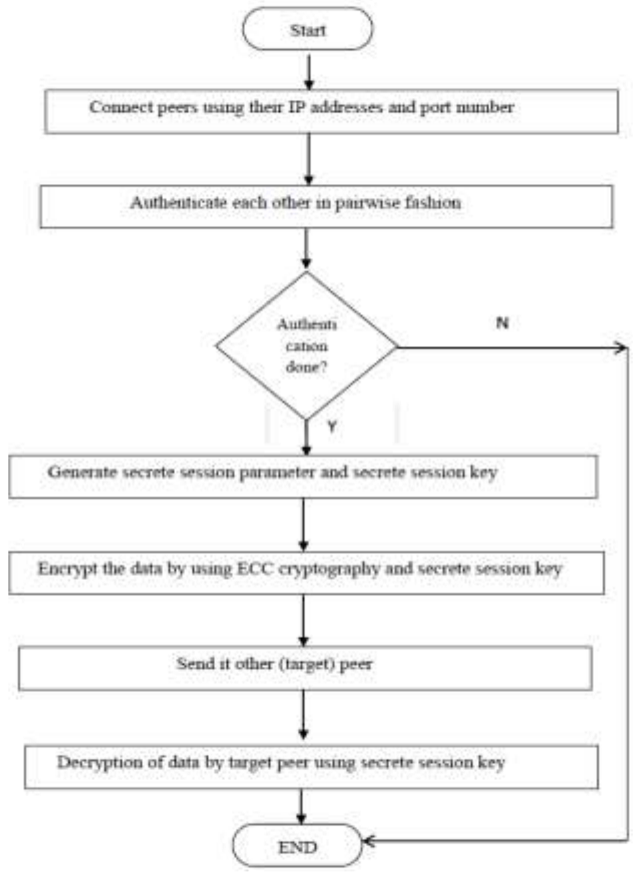


Fig: 3.2. Flow graph working of proposed system

Cryptographic Primitives

In this section, we describe some basic cryptographic primitives which are useful to implement and understand our proposed protocol.

Let G1 be an additive group and G2 be a multiplicative group of the same prime order q. Let P be an arbitrary generator of G1. Note that aP denotes P added to itself a times. Assume that the discrete logarithm (DL) problem is hard in both G1 and G2. We can think of G1 as a group of points on an elliptic curve over Fq, and G2 as a subgroup of the multiplicative group of a finite field Fq k for some k ∈ Zq*, where Zq*={ξ| 1 ≤ ξ ≤ q−1 }. A mapping e:G1 ×G1→ G2, satisfying the following properties, is called a cryptographic bilinear map.

 Bilinearity: e(aP, bQ)=e(P,Q)ab=e(bP, aQ) ∈ G2 for all P,Q ∈ G1 and a, b ∈ Zq*. This can be restated in the following way. For all P,Q,R ∈ G1; then e(P+Q, R) = e(P,R) e(Q,R) = e(Q,R) e(P,R) ∈ G2 and e(P, Q+R) = e(P,Q) e(P,R) = e(P,R) e(P,Q) ∈ G2.

 Non-degeneracy: If P is a generator of G1, then e(P,P) is a generator of G2. In other words, e(P,P) ≠ 1.

 Computable: A mapping is efficiently computable if e(P,Q) can be computed in polynomial-time for all P, Q ∈ G1. Modified Weil Pairing [3] is an example of cryptographic bilinear map.

Let the group G1 represents the group of points on the elliptic curve E:Y2=X3+ αX+ β mod τ, where τ is a prime number, then using the group G1, we can define the following hard cryptographic problems applicable to our proposed protocol.

 Computational Diffie-Hellman (CDH) Problem: Given a triple (P, aP, bP) ∈ G1 for a, b ∈ Zq*, find if there exists any element abP ∈ E.

 Decisional Diffie-Hellman (DDH) problem: Given a quadruple (P, aP, bP, cP) ∈ G1 for a, b, c ∈ Zq*, decide whether c=ab mod q or not.

 Gap Diffie-Hellman (GDH) Problem: A class of problems where the CDH problem is hard but the DDH problem is easy.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

Data Exchange Setup For P2DSS

In this section, we introduce and study a framework, called peer data exchange, which is a generalization of data exchange and a special case of a full-fledged peer data management system. This framework models a situation in which there is interaction between two peers that have different roles and capabilities: one of them, called the source peer, is an "authoritative" or "trusted" peer that can contribute new data, while the other peer, called the target peer, imposes restrictions on the data that it is willing to accept, but has no permission or capability to modify the data of the source peer. In a peer data exchange setting, the relationship between the two peers is specified by constraints that go in either direction, that is, some are source-to-target constraints and others are target-to-source constraints; in addition, target constraints may be present. As in data exchange, the source-to-target constraints specify what data a source peer is willing to exchange.

Attributes are symbols taken from a given finite set U={A1,…,Aq} called the universe. We use the letters A, B, C, … to denote single attributes and X, Y, … to denote sets of attributes. Each attribute Aj is associated with a finite set of values called the domain of Aj and is denoted by dom(Aj). Suppose X={A1,A2,…,Ak} $\subseteq$ U, with the elements Ai($1 \le i \le k$) taken in the order shown,

then dom(X) $\subseteq$ dom(A1)×dom(A2)×…×dom(Ak). A non-empty subset of U is called a relation schema R. A database schema is a finite collection $\mathfrak{R}$ = (R1,…,Rm) of relation schemas.

Let S be a schema at a peer Pi and T be a schema at another peer Pj. If a data exchange policy is specified from S to T, then we call S a source schema and T a target schema. Each peer has instances corresponding to its schema. Next we discuss the data exchange settings. Generally, in data exchange settings [7].

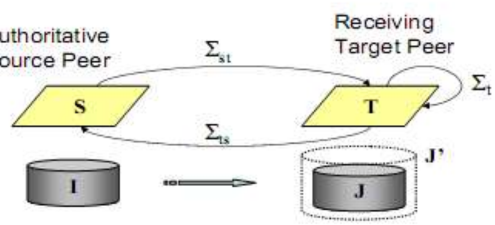


Fig2. Illustration of Peer Data Exchange

source-to-target data exchange policies are constituted by a set of assertions of the forms

$\Sigma st = qS \rightarrow qT$

where, qS and qT are two queries, respectively over the source schema S, and over the target schema T. Intuitively, an assertion qS→ qT specifies that the concept represented by the query qS over the sources corresponds to the concept in the target schema represented by the query qT. The assertions are basically tuple-generating dependencies [8]. Assertions can be specified as logical expressions of the form:

$\forall x[\exists w \phi(x,w) \rightarrow \exists z \psi(x,z)]$

where, the left-hand side (LHS) of the implication, ϕ, is a conjunction of relation atoms over the schema of S and the right-hand side (RHS) of the implication ψ is a conjunction of relation atoms over the schema T. The policy expresses a constraint about the appearance of a tuple in the instance satisfying the constraint of the RHS, given a particular combination of tuples satisfying the constraint of the LHS. Basically, the policies provide a structural relationship of data between source and target as well as allowing data to be exchanged between the two. Through the policies, a source also exports part of its schema accessible to the target. The following is a simple example of a data exchange setting.

The shared attributes, confidential attributes and non confidential attributes can defined as follows:

Shared attributes: Consider two peers Pi and Pj in a P2PDBS. Let S be a schema with a set of attributes Us in Pi and T be a schema with a set of attributes Ut in Pj. Assume a policy Σst=qS→ qT between Pi and Pj. Let att(Σst) denote the set of attributes exposed by Pi using the policy Σst. Therefore, the shared attributes, denoted by SA, are SA $\subseteq$ Us = att(Σst).

Confidential attributes: Consider a data sharing policy between two peers Pi and Pj is Σst=qS→ qT.

Let SA be the set of shared attributes. Therefore, the confidential attributes, denoted by CA, are CA $\subseteq$ SA.

Non-confidential attributes: Consider a data sharing policy between two peers Pi and Pj is Σst=qS→ qT. Let SA be the set of shared attributes and CA be the set of confidential attributes. Hence, the non-confidential attributes, denoted by NCA, are SA − CA.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

Private attributes: Consider the data sharing policy $\Sigma st=qS\rightarrow qT$ between two peers Pi and Pj and let SA be the set of shared attributes, the private attributes, denoted by PA, is Us − SA.

IMPLEMENTATION OF PROPOSED SYSTEM

Transferring information between peers. A peer Pi may request a transfer of information from a peer Pj, by sending a transfer request message to Pj. Pj, upon receiving this message checks whether it has the information item associated with the request. If Pj has the item then Pj transfers the requested information to Pi. If the information is transferred to Pi, then Pi becomes the owner of that copy of the information. The security requirements for information transfer are:

1. The transfer request message and the transfer of the information are confidential between Pi and Pj.
2. Pi and Pj are able to identify each other and thus determine the level of their trust relationship.
3. The information is transferred from Pj to Pi only if Pi is authorized to access that information.

To prevent the attacks, an "on-the-fly" security setup is needed between the source Pi and the target Pj, based on the query. Assume a source peer Pi with schema S and a target peer Pj with schema T. Also assume that based on the data exchange policy between Pi and Pj the shared attributes are classified as follows:

Confidential attributes (CA) = {CA1,CA2,…,CAm }

Non-confidential attributes (NCA) = {NCA1, NCA2,…,NCAp }

The purpose of the security protocol is to ensure secure data exchange when Pj requests data from Pi through a query Q that contains confidential attributes as well as non-confidential attributes. Assume a query Qt at any time instance t is requested from Pj to Pi. Before forwarding the query Qt, Pj generates system as well as session parameters.

System parameters:

System parameters (e.g. group, bilinear map, hash function) are used for generating secret session keys for data exchange between peers. Depending on the mutual agreement between peers, system parameters may be fixed for each data exchange session or they may be changed for each session.

Session parameters:

Session parameters (e.g. dynamically generated id of peers, random number in Zq*, random numbers) are used for a specific data exchange session in order to generate the secret session key. These parameters are dynamic for each session of data exchange. In order to request data from Pi, peer Pj generates the following system and session parameters.

System parameters:

G1, an additive group of prime order q.

- H1:{0,1}*→ G1, a collision resistant cryptographic hash function which maps from arbitrary-length strings to points in G1.

Session parameters:

- IDPj = H1(Pjγ) ∈ G1 , a dynamically generated id of peer Pj, where γ is a random number. After creating the parameters < G1, H1, IDPj > , peer Pj sends the parameters with the query Qt to Pi. When Pi receives the parameters and the query, it identifies the confidential and non-confidential attributes. Assume Pi identifies the following confidential and non-confidential attributes from the query Qt: Confidential attributes in Qt, denoted by CAQt={QCA1,QCA2,…,QCAm } ⊆ CA Non-confidential attributes in Qt, denoted by NCAQt = {QNCA1,QNCA2,…,QNCAp } ⊆ NCA When Pi receives the parameters from Pj, it also generates system and session parameters for computing a secret session key for the authentication of Pj and for encryption of the query result, Qt R. The generated parameters are given below.

System parameters:

- G2, a multiplicative group of the same prime order q as the order of the additive group G1.
- A bilinear map ~e:G1 ×G1→ G2.
- H2, H3, two collision resistant cryptographic hash functions. H2:{0,1}n−k ×{0,1}k → Zq*, where

Zq* = {μ| 1 ≤ μ ≤ q−1 }. H3:{0,1}* → {0,1}λ; a mapping from arbitrary-length strings to λ -bit fixed length string.

Session parameters:

- An ID IDPi = H1(Piζ) ∈ G1 , where, ζ is a random number.
- A random number Ri−SESSION which is used for generating the authentication code Aut0. Depending on the confidential and non-confidential attributes, Pi now generates the secret session key KSi and authentication code Aut0 using its own parameters and the parameters of Pj. The generation and purpose of KSi and Aut0 are discussed as follows:

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

Secret Session Key and Authentication Code

In identity-based crypto there is generally a private key generator (PKG) which entities use in order to obtain their private keys. This is a trusted authority (like a CA in a PKI). In our proposed protocol there is no PKG but still our protocol works properly. In this proposed security protocol, the responsibilities of a PKG are mutually performed by the source and the target. The source Pi computes a shared secret element in Zq*, called a shared secret parameter and denoted as σ based on the query attribute sets CAQt and NCAQt as follows:

σ = H2(NCAQt ×CAQt) ∈ Zq*Pi also computes another shared secret identity in G1, called shared secret identity, denoted by IDSP based on the query attribute set CAQt as follows:

IDSP = H1 (CAQt) ∈ G1

Depending on the query attributes, session key KSi for each session is generated by the source Pi as follows:

KSi =~e(IDPi+ IDPj, σIDSP)

=~e(IDPi, σIDSP)~e(IDPj, σIDSP)

=~e(IDPi, IDSP)σ ~e(IDPj, IDSP)σ

Source Pi also generates authentication code Aut0 as follows:

Aut0 = H3(KSi|| IDPi || IDPj || Ri−SESSION || 0) where Ri−SESSION is a random number generated by the source Pi to distinguish every session from each other so that a replay attack cannot take place on the communication. Finally, source Pi sends the system parameters < G2, ~e, H2, H3 > including the session parameters < IDPi, Ri−SESSION, Aut0 > to the target Pj. After receiving the system parameters as well as session parameters from the source Pi, target Pj generates σ and IDSP. Finally target Pj computes a session key KSj as follows:

KSj =~e(IDPj+ IDPi, σIDSP)

=~e(IDPj, σIDSP)~e(IDPi, σIDSP)

=~e(IDPj, IDSP)σ ~e(IDPi, IDSP)σ

=~e(IDPi, IDSP)σ ~e(IDPj, IDSP)σ

=KSi

Target also computes the verification code Ver0 as follows:

Ver0 = H3(KSj|| IDPi|| IDPj|| Ri−SESSION ||0) The verification code Ver0 is computed to verify the authentication code Aut0 of Pi. Target Pj compares Ver0 with Aut0; if (Ver0 = Aut0) then target generates another authentication code Aut1 as follows:

Aut1 = H3(KSj|| IDPi|| IDPj|| Rj−SESSION|| Ri−SESSION || 1) where Rj−SESSION is a random number generated by the target and different from each session so that replay attack (request to source) cannot take place in the communication. Finally, Pj sends < Aut1, Rj−SESSION > to source Pi. Upon receiving < Aut1, Rj−SESSION > from the target Pj, source Pi generates another verification code Ver1 as follows, and compares it with Aut1.

Ver1 = H3(KSi|| IDPi|| IDPj|| Rj−SESSION|| Ri−SESSION || 1) If Ver1 matches Aut1 , i.e (Ver1 = Aut1) then source peer sends the data of the query result Qt R by encrypting it with the private session key KSi. For distinguishing the computation of authentication codes by the source and the target and the communication of the authentication codes between the source and the target, "0" and "1"are used.

Secure Authenticated Data Exchange

After authentication between the source and the target, source Pi generates a message authentication code, denoted by MACMESSAGE on query result Qt R, which is computed as MACMESSAGE = H3(Qt R). The source also encrypts QtR with its secret session key KSi, denoted by CIPHER Qt R, which is computed as CIPHER QtR = EKSi(Qt R), where EKSi means encryption using the session key KSi. Finally, Pi sends the following packet to Pj. < IDPi, CIPHERQtR, MACMESSAGE,IDPj > After receiving the packet, Pj decrypts CIPHERQtR with the session key KSj denoted as DKSj(CIPHERQtR) and generates the verification message authentication code, denoted by VERMESSAGE, which is computed as follows:

VERMESSAGE=H3(DKSj(CIPHERQtR))

Finally, Pj compares VERMESSAGE with MACMESSAGE. If VERMESSAGE = MACMESSAGE then the data is accepted. .

The step-by-step procedure of the proposed protocol :

STP 1: A query Qt is generated at the target Pj.

STP 2: Target Pj determines group G1, hash function H1 and performs the following steps:

2.a: Generates an ID IDPj ;

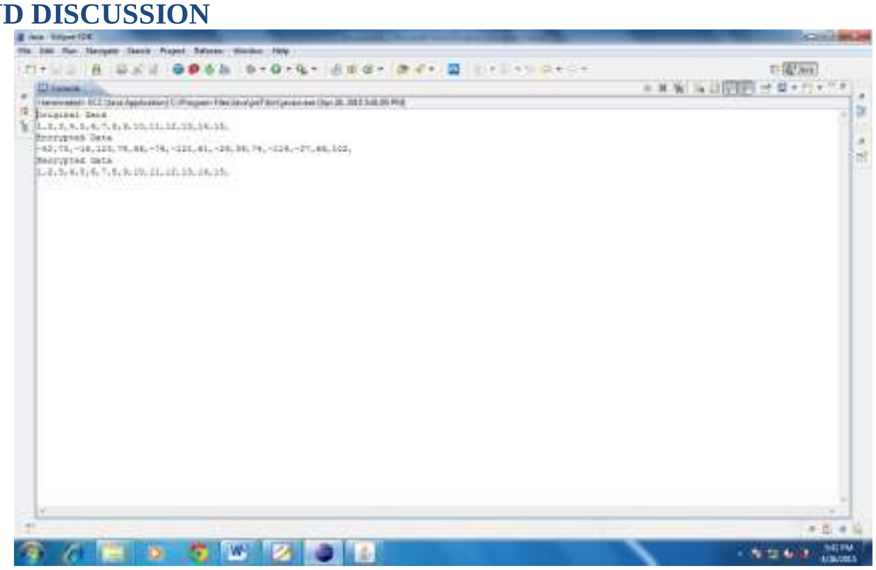
2.b: Sends < G1, H1, Qt, IDPj > to the source Pi.

STP 3: Source Pi executes the query Qt on its local database and performs the following steps:

3.a: Determines group G2, bilinear mapping function ~e, and cryptographic hash functions H2 and H3.

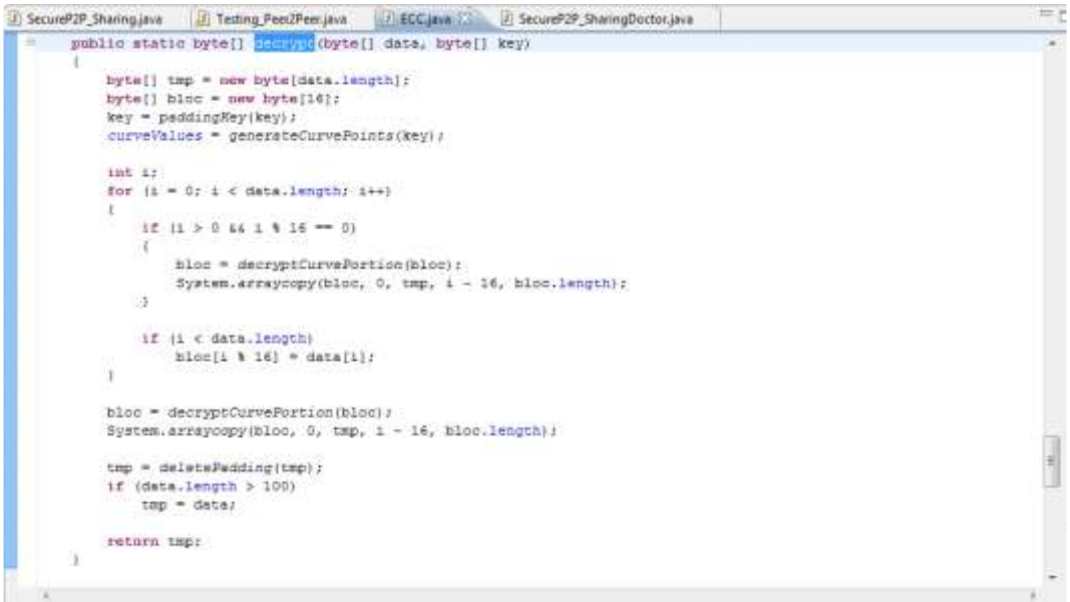
3.b: Generates an ID IDPi, a random number Ri−SESSION.
3.c: Generates secret session key KSi, authentication code Aut0.
3.d:Sends<G2,~e,H2,H3,IDPi,Ri−SESSION,Aut0> to Pj.
STP 4: Target Pj generates session key KSj, verification code Ver0.
4.a: Generates Rj−SESSION; and Compares Ver0 with Aut0; if Ver0 = Aut0 then generates Aut1.
4.b: Sends < Rj−SESSION, Aut1 > to the source Pi.
STP 5: Source Pi generates verification code Ver1.
5.a: Compares Ver1 with Aut1; if Ver1 = Aut1 then generates message authentication code MACMESSAGE.
5.b: Encrypts query result Qt R, with the session key KSi, denoted as CIPHERQtR;
5.c: Sends < IDPi,CIPHERQtR, MACMESSAGE, IDPj > to the target Pj.
STP 6: Target decrypts CIPHERQtR with session key KSj; generates verification message authentication code VERMESSAGE; compares VERMESSAGE with MACMESSAGE; if VERMESSAGE = MACMESSAGE then data is exchanged successfully.

RESULTS AND DISCUSSION

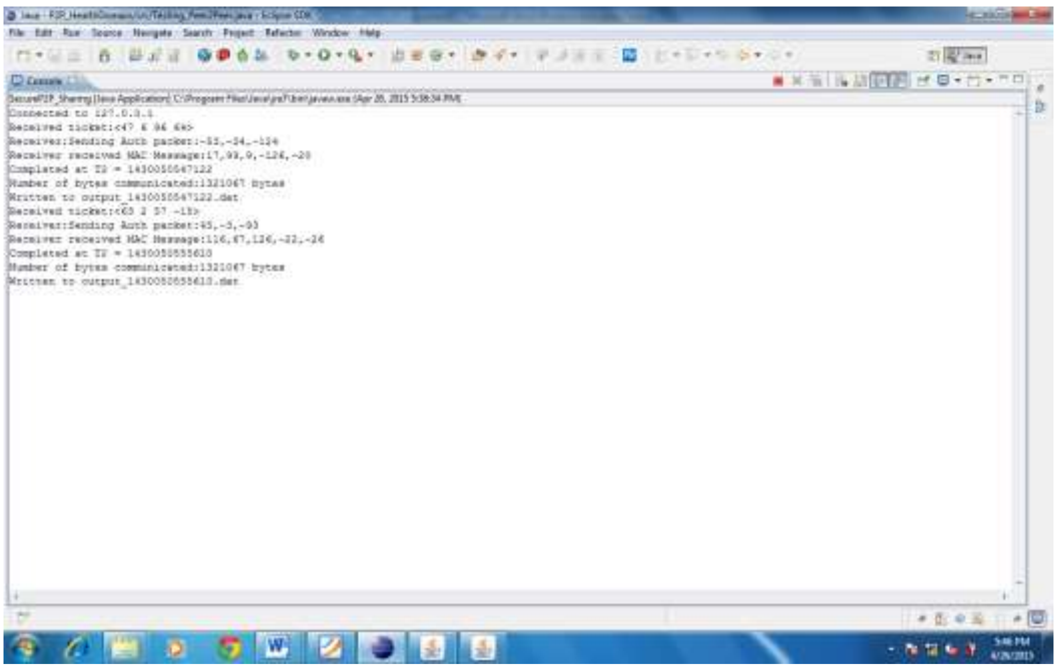


Scr1:ECC demonstration for encryption

Scr2: Encryption of data with secrete session key



Scr3: Decryption of data at target peer



Scr4: Sharing of data from one peer to anather

Attack analysis

In our protocol the secret keys KSi and KSj are generated based on the confidential and the non-confidential attributes that are only shared between the source and the target peers. Therefore, an intruder node cannot generate a session key in the middle of a data exchange session between two peers. Thus, man-in-the-middle attack is not effective on the proposed protocol .A masquerade may be attempted through the use of stolen logon IDs and passwords, through finding security gaps in programs, or through bypassing the authentication mechanism. In this proposed protocol, peers authenticate each other before exchanging data. Furthermore, in every session of data exchange between peers, parameters (session/system) are generated dynamically. The session parameters <Ri−SESSION, Aut0, Aut1, Rj−SESSION > are completely different in each session. Hence, by storing these session parameters and using these parameters in challenge/response session during authentication phase, an intruder node cannot pass the authentication process. Therefore, the intruder cannot pretend to be a valid peer in the data exchange. Thus, a masquerade attack is prevented.

CONCLUSION

We have implemented Secure data exchange protocol for P2PDSS in public health domain using pairing-based cryptography and ECC for encryption in java. Using this technique any two peer can communicate over insecure medium by generating new session key for each data exchange session making every session independent of previous which helps to avoid man in middle attack and Masquerade Attack and reply attack.

ACKNOWLEDGEMENTS

The authors would like to thank Yogesh Yerande for many fruitful discussions about P2P network security and his helpful critiques of this paper.

REFERENCES

1. N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, Vol.48, pp. 203-209, 1987.
2. L. B. Oliveira, R. Dahab, "Pairing-Based Cryptography for Sensor Networks," 5th IEEE International Symposium on Network Computing and Applications (NCA'06), MA, USA, Jul. 2006
3. MehediMasud and Sk. Md. Mizanur Rahman "Secure Data Exchange in P2P Data Sharing Systems in eHealth Perspective" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012
4. D. Boneh and M. Franklin, "Identity-based Encryption from the Weil Pairing," Proc. CRYPTO 2001, LNCS 2139, Springer-Verlag, Berlin Heidelberg, Santa Barbara, CA, USA, pp. 213-229, 2001.
5. N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs," Workshop on Cryptographic Hardware and Embedded Systems (CHES-2004), Cambridge, MA, USA, pp. 119-132, 2004.
6. Bogdan C. Popescu, Bruno Crispo, Andrew S. Tanenbaum, "Safe and Private Data Sharing with Turtle: Friends Team-Up and Beat the System," Lecture Notes in Computer Science (LNCS 3957), pp. 213-220, 2006.
7. Sk. Md. M. Rahman, M. Masud, C. Adams, H. T. Mouftah and A. Inomataz, "Session-wise private data exchange in eHealth peer-to-peer database management systems,", IEEE International Conference on Intelligence and Security Informatics (ISI), 2011
8. A. Fuxman, P. G. Kolaitis, R. J. Miller, and W. C. Tan, "Peer Data Exchange," ACM Trans. Database System, Vol. 31, Issue. 4, pp.1454-1498, 2005.
9. C. Beeri and M. Y. Vardi, "A Proof Procedure for Data Dependencies," Journal of the ACM, Vol. 31, Issue. 4, pp. 718-741, 1984.
10. H. W. Lim, "On the Application of Identity-Based Cryptography in Grid Security," Ph.D thesis, University of London, 2006
11. P. Barreto, S. Galbraith, C. O. hEigeartaigh, and M. Scott, "Efficient pairing computation on supersingular Abelian varieties," Designs, Codes and Cryptography, Vol. 42, pp. 239-271, 2007.
12. A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov, "Schema Mediation in Peer Data Management System" , Proc. of the Int'l Conf. on Data Engineering, pp. 505-516, 2003.
13. P. Rodriguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R. Miller, and J. Mylopoulos, "Data Sharing in the Hyperion Peer Database System," Proc. of the Int'l Conf. on Very Large Data Bases (VLDB), pp. 1291-1294, 2005.
14. A. Kementsietsidis, M. Arenas, and R.J. Miller, "Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues," Proc. of the Int'l Conf. on the Management of Data (ACMSIGMOD), pp. 325-336, 2003.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

AUTHORS BIBLIOGRAPHY

	**Amol G. Kadu** Pursuing Master of Engineering in Computer Engineering from Sipna's College of Engineering and Technology, Amravati
	**Dhananjay M. Dakhane** Received the Master of Engineering in CSE, pursuing P.H.D from SGB Amravati University. Working as an Associate Professor at Sipna's College of Engineering and Technology. Amravati.